First Named Inventor	Rolia et al.
Serial No.	10/804,724
Filing Date	03/19/2004
Group Art Unit	2145
Examiner Name	Williams, Clayton R.
Confirmation No.	8258
Docket No.	200300271-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Rolia et al.

Title: COMPUTING UTILITY POLICING SYSTEM AND METHOD USING ENTITLEMENT PROFILES

<div align="right"><u>VIA EFS</u></div>

<div align="center"><u>APPEAL BRIEF</u></div>

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

As required under 37 C.F.R. § 41.37(a), this Appeal Brief is being submitted in furtherance of the Notice of Appeal filed on August 05, 2008. The fees required under 37 C.F.R. § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF. This Appeal Brief contains items under the following headings:

<div align="center"><u>TABLE OF CONTENTS</u></div>

I.   REAL PARTY IN INTEREST

The present application has been assigned in an assignment recorded March 19, 2004 , at Reel 015123, Frame 0886 to Hewlett-Packard Development Company, L.P., a Limited Partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

II.   RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any other related appeals or interferences that may directly affect or be directly affected by or have a bearing on the Board's decision in the present appeal.

III.   STATUS OF THE CLAIMS

Claims 1, 15, 29 and 30 stand rejected under 35 U.S.C. § 103(a) over U.S. Patent No. 6,877, 035 to *Shahabuddin et al.* (hereinafter "Shaha") and in view of U.S. Patent 5,274,644 B1 to *Berger* et al. (hereinafter "Berger") and further in view of U.S. Patent Publication 2005/0165925 A1 to *Dan et al.* (hereinafter "Dan").

Claims 2-12 and 16-26 stand rejected under 35 U.S.C. § 103(a) over Shaha in view of Berger.

Claims 14 and 28 stand rejected under 35 U.S.C. § 103(a) over Shaha, as applied to Claims 1 and 15 in view of Berger and further in view of Dan.

Claims 13 and 27 are cancelled.

Appellant appeals the rejection of the aforementioned pending claims 1-12, 14-26, and 28-30 which are set forth in the attached Appendix A.

IV.   STATUS OF AMENDMENTS

The amendments to the claims have all been entered and Appellant is unaware of any amendments filed after the Final Office Action mailed 05/02/2008 which finally rejected claims 1-12, 14-26, and 28-30.

V.   SUMMARY OF CLAIMED SUBJECT MATTER

Claims 1-12, 14-26, and 28-30 are directed to a system and method for policing resources in a computing utility facility.

Representative claim 1 describes a method that intercepts an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application. The method acquires an entitlement profile associated with the application to determine if application is entitled to requested resources over a time period. Using the entitlement profile, the method identifies an entitlement value and a corresponding sliding window of the time period. The method then determines if the advance request for resources from the application exceeds the entitlement value associated with the sliding window. In response to the determination, the method indicates the application's entitlement to the request for resources in response to the determination. If the advance request from the application is excessive then the method includes a throttling of the requested resources in accordance with the entitlement profile as the application is not

entitled to the additional resources. (Appellant's Specification, FIG. 1, FIG.7, FIG. 8,

FIG. 9, paragraphs [0011], [0014] [0068]-[0077]).)

Representative Claim 15 is directed to an apparatus for policing resources in a

computing utility facility. Representative claim 15 describes an apparatus that intercepts

an advance request for resources from an application admitted to access a pool of

resources associated with the computing utility facility and prior to utilization of the pool

of resources to execute the application. The apparatus acquires an entitlement profile

associated with the application to determine if the application is entitled to requested

resources over a time period. Using the entitlement profile, the apparatus identifies an

entitlement value and a corresponding sliding window of the time period. The method

then determines if the advance request for resources from the application exceeds the

entitlement value associated with the sliding window. In response to the determination,

the method indicates the application's entitlement to the request for resources in response

to the determination. If the request is excessive includes a throttling of the requested

resources when the application is not entitled to the additional resources in accordance

with the entitlement profile (Appellant's Specification, FIG. 1, FIG.7, FIG. 8, FIG. 9,

paragraphs [0011], [0014] [0068]-[0077]).

Representative Claim 29 is directed to a computer program product with

instructions executed on a processor that polices resources in a computing utility facility.

Representative claim 29 describes instructions that intercept an advance request for

resources from an application admitted to access a pool of resources associated with the

computing utility facility and prior to utilization of the pool of resources to execute the

application. The instructions acquire an entitlement profile associated with the

application to determine if the application is entitled to requested resources over a time period. Using the entitlement profile, the instructions identify an entitlement value and a corresponding sliding window of the time period. The instructions then determine if the request for resources from the application exceeds the entitlement value associated with the sliding window. In response to the determination, the instructions indicate the application's entitlement to the request for resources in response to the determination. If the advance request is excessive then the instructions include a throttling of the requested resources as the application is not entitled to the additional resources. (Appellant's Specification, FIG. 1, FIG.7, FIG. 8, FIG. 9, paragraphs [0011], [0014] [0068]-[0077]).)

Representative Claim 30 is directed to an apparatus for policing resources in a computing utility facility. Claim 30 includes a means for intercepting an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application. The apparatus further includes a means for acquiring an entitlement profile associated with the application to determine if application is entitled to requested resources over a time period. Additionally, Claim 30 includes a means for identifying an entitlement value and corresponding sliding window of the time period from the entitlement profile. The apparatus also includes a means for determining if the request for resources exceeds the entitlement value associated with the sliding window. A means for indicating application entitlement to the request for resources in response to the determining is also included. If the request is excessive, then the apparatus includes a throttling of the requested resources when the application is not entitled to the additional

resources in accordance with the entitlement profile. (Appellant's Specification, FIG. 1,

FIG.7, FIG. 8, FIG. 9, paragraphs [0011], [0014] [0068]-[0077]).)

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

    A.   WHETHER CLAIMS **1, 15, 29** AND **30** ARE NONOBVIOUS OVER SHAHA IN VIEW OF BERGER AND FURTHER IN VIEW OF DAN.

    B.   WHETHER CLAIMS **2-12** AND **16-26** ARE NONOBVIOUS OVER SHAHA IN VIEW OF BERGER.

    C.   WHETHER DEPENDENT CLAIMS **14** AND **28** ARE NONOBVIOUS OVER SHAHA AS APPLIED TO INDEPENDENT CLAIMS **1** AND **15,** IN VIEW OF BERGER AND FURTHER IN VIEW OF DAN.

VII.   ARGUMENT

    Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board overturn the outstanding rejections in light of the remarks contained herein. The claims do not stand or fall together. In fact, Appellant presents separate arguments for various independent and dependent claims. Each of these arguments is presented with separate headings and sub-headings, as required by 37 C.F.R. § 41.37(c)(1)(vii).

    A.   CLAIMS **1, 15, 29** AND **30** ARE NONOBVIOUS OVER SHAHA IN VIEW OF BERGER AND FURTHER IN VIEW OF DAN.

The Shaha Reference

    Shaha concerns one method of allocating hosting-services on a shared server to one or more clients. (Abstract) Shaha's the basic allocation method includes two steps:

        (1) discovering a utilization pattern of host-service resources from clients by monitoring the clients access rates of one or more websites. (Shaha at Col. 2, lines 59-63; Col. 3, lines 16-18; Col. 3 lines 58-61; Col. 4, lines 12-16; Col. 6, lines 10-15; Col. 7, lines 40-42)

(2) allocating the host-service resources to the clients dependent on the utilization patterns. (Shaha at Col. 2, lines 64-65; Col. 3, lines 22-30; Col. 3, lines 62-64; Col. 4, lines l6-18)

Indeed, Shaha mentions certain embellishments to these two fundamental steps but the general approach remains the same. For example, Shaha mentions that the "discovering" in step (1) may include dividing a day into 24 1-hour slots and then modeling a clients need for resources as clients access different "web sites" during each hour of the day. (Shaa at Col. 6, lines 9-27) With respect to the "allocating" step (2), Shaha mentions that distribution of resources may also depend on attempting to provide a certain minimum Quality of Service (QoS) particular to each client. (Shara at Col. 3, lines 14-15; Col. 3, lines 62-64; Col. 4, lines 16-17; Col. 5, lines 14-20) According to Shaha, the QoS is expressed in a contractual Service Level Agreement (SLA) negotiated by each client to provide a minimum level of service. (Shaha at Col. 6, lines 28-36; Col. 7, lines 49-53) Appellants submit that the SLA in Shaha is not a request for specific resources. (Shara at Col. 6, lines 32-34) Instead, it is a contractual agreement and guarantee that a client will pay a minimum and maximum for delivering a range of resources over time. (Shara at Col. 5, lines 25-28; Col. 6, lines 32-34) Accordingly, the SLA merely determines the cost associated with using certain resources but it does not operate as a request by the client for resources.

It is important to also note that a "client" in Shaha is not an application but a computer capable of running one or more applications. (Shaha at Col. 5, lines 38-41) Shaha allocates resources to specific combinations of client computers and not to specific

individual applications. Shaha relies on allocating resources to combinations of computers especially if they are in different time zones or exhibit other complementary peak periods in disjoint intervals. (Shaha at Col. 5, lines 45-48) For example, it is important for Shaha's operation to allocate resources to combinations of computers located in different time zones (i.e., India and the United States) in order to balance out the demand for resources during a given time interval. (Shaha at Col. 5, lines 41-44) Essentially, the "complementarity" requirement in Shaha dictates that resources are not granted to one client but to a combination of clients that share resources. Nothing in Shaha provides any method of determining what resources are being used by an application or how certain resources could be delivered to a certain application as this would go against a central design feature of Shaha: complementarity. (Shara at Col. 5, lines 55-67; Col. 6, lines 1-8)

The Berger Reference

Berger presents a method of sharing a common resource over several different classes. (Abstract) In Berger, each of N different classes are assigned a token bank i (bank 'i' from i = 1 to N) that holds a finite number of "tokens". (Berger at Col. 3, lines 21-61) The token bank is essentially implemented as a counter for each class to keep track of the tokens for that class. (Berger at Col. 3, lines 26-29) A spare or extra bank (bank 'N+1') is provided in Berger in addition to the individual token bank 'i' to represent the excess shared resources not assigned to any one class.

Tokens allow requesters from each class to access a shared resource as long as there are tokens available in the respective bank for the particular class. Thus, the requester gains access to a particular resource by first using tokens in the requester's

token bank (Berger at Col. 3, lines 38-41) If the requester's token bank is empty then the requester may use a token from a common or extra bank (bank 'N+1'). (*Id.*) When there are no tokens available in the extra bank or the token bank for the class then access to the resource is denied. (Berger at Col. 3, lines 41-43)

Likewise, incoming tokens are first assigned to the token bank for the particular class until the token bank for the class becomes full. Tokens that encounter a full token bank for a particular class may then be credited to the common bank of tokens (bank 'N+1'). When the common bank (N+1) becomes full then any additional tokens are lost as they cannot be credited to either the particular token bank for a class or the common bank of tokens. (Berger at Col. 3, lines 56-61)

Tokens arrive and depart from the different token banks at preassigned rates associated with the class 'i' and according to the function "rate(i)". (Berger at Col. 3, lines 52-56) The rate(i) is not requested by the client but instead is setup in advance as a guaranteed or contracted admission rate for each class 'i'. (Berger at Col. 6, lines 1-8) Berger defines rate(i) as a deterministic flow rate of tokens and therefore credit associated with a class i. (Berger at Col. 8, lines 14-19) Indeed, it is important that clients do not request or change the rate(i) over time as this would make the system non-deterministic and contradict the express definitions and organization in Berger. Essentially, the rate(i) is fixed for different classes in advance. (Berger at Col. 11, lines 39-42)

Berger uses the term 'throttle' or switch to indicate whether a class will be able to access a resource. (Berger at Col. 5, lines 21-21-25) Every requester consumes a token to obtain access to the resources. If a class 'i' request comes in then the requester first

checks if there is a token in the token bank associated with requestor i. When there is no token in i^{th} token bank and no further token's exist in the common bank then the 'throttle' operates to disable or close access to the resources. Conversely, if there is a token in the token bank 'i' or even in the common bank then the 'throttle' opens or enables the requester to access resources. As described in the specification, the 'throttle' specifically operates as a switch to either enable access to resources or disable access to resources. (Berger at Col. 6, lines 22-32)

The Dan Reference

Dan concerns a distributed data processing system for allocating computing resources on different domains based on a workload and a service level agreement (SLA). In operation, Dan monitors a current workload on a data processing system and then attempts to predict a future workload on the system. (Dan at Abstract and paragraph 0013) Dan measures transaction rates, response times and other metrics as transactions are being executed. (*Id.*) If there is a sudden increase in the predicted workload then Dan may request and receive resources from remote sites and domains to handle the expected increase in processing. (Dan at paragraph 0019)

The SLA determines what cost is associated with executing a workload. This cost is included along with a request made to the remote sites for additional resources when the current resources are not sufficient. (Dan at paragraph 0019) Essentially, the service level agreement (SLA) is a legal contract and abstract concept that determines the overall computing cost. It is neither a direct or indirect measure of the computing resources but a wishful list of performance requirements and the penalties one side or the other shall pay in the event they do not abide by the contract. (Dan at paragraphs 0093-0098) Moreover,

the SLA in Dan does not operate as an advance request for resources since the resources required are being measured and then predicted.

        a. ALL CLAIM LIMITATIONS ARE NOT TAUGHT OR SUGGESTED BY THE CITED ART.

As indicated below, it is respectively submitted that the Examiner has not established a prima facie case of obviousness or satisfied the all elements rule as required by 35 U.S.C § 103(a), MPEP § 2143.03 and the relevant case law.

Regarding claim 1, the combination of Shaha, Berger and Dan do not teach or suggest "intercepting an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application." With regards to the prima facie case, it is a requirement that all the claim limitations must be taught or suggested by the prior art. MPEP § 2143.03 In re Royka , 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson , 424 F.2d 1382, 165 USPQ 494, 496 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. In re Fine , 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

Appellants submit that Shaha does not operate by "intercepting an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application" as recited in Claim 1. This is because there are no requests for resources made in advance by applications and prior to their execution in

Shaha. Instead, Shaha simply operates by monitoring the activity of various clients *as they execute* and not in advance. Specifically, Shaha operates by "discovering a utilization pattern of host-service resources from clients by monitoring the clients access rates of one or more websites" (Shaha at Col. 2, lines 59-63; Col. 3, lines 16-18; Col. 3 lines 58-61; Col. 4, lines 12-16; Col. 6, lines 10-15; Col. 7, lines 40-42) For at least this reason, Appellants respectfully request that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to independent Claim 1.

Appellants also submit that Shaha should not be modified to include the limitation in Claim 1 of "intercepting an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application" as suggested by the Examiner. (Final Office Action May 2, 2008, page 4, lines 4-9) The Examiner admits that Shaha lacks this limitation but provides no suggestion for the desirability of such a modification in the cited art. (Final Office Action May 2, 2008, page 3, lines 14-18) *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984) ("The fact that a prior art device could be modified so as to produce the claimed device is not a basis for an obviousness rejection unless the prior art suggested the desirability of such a modification")

For example, the Examiner's argument that Dan "teaches that sharing resources effectively can result in better overall utilization of computing resources" (Dan, [0009], lines 3-5) does not suggest the desirability of having an application make an advance request for resources before it executes. Indeed, SLAs in Dan are merely contractual agreements concerning pricing but do not teach or suggest making "an advance request for resources from an application" as recited in Claim 1. Overall, suggestions made in

Dan are too general and not specific enough for one skilled in the art to successfully modify Shaha and arrive at the method recited in Claim 1. For at least this additional reason, Appellants respectfully request that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to independent Claims 1.

Furthermore, Shaha teaches away from receiving advance requests from a single application as this goes against the concept of "complementarity" as defined and used in Shaha. (Shaha at Col. 5, lines 45-54) Teaching away from receiving advance requests from an application negates any possible suggestion or motivation to modify the reference. *Ormco Corp. v Align Technologies, Inc.* 463 F.3d 1299 (Fed. Cir. 2006) ("[A] reference that 'teaches away' from a given combination may negate a motivation to modify the prior art to meet the claimed invention.") Complementarity is defined in Shaha as allocating resources to client combinations and "choosing such client combinations...to determine clients that have peak periods in almost disjoint intervals." (Shaha at Col. 5, lines 45-54)

Shaha relies on allocating resources to different computers especially if they are in different time zones or exhibit other complementary peak periods in disjoint intervals. (Shaha at Col. 5, lines 45-48) For example, it is important for Shaha's operation to allocate resources to combinations of computers located in different time zones (i.e., India and the United States) in order to balance out the demand for resources during a given time interval. (Shaha at Col. 5, lines 41-44) Essentially, the "complementarity" requirement in Shaha dictates that resources are granted to combination of clients that implicitly share the resources. For at least this reason, Shaha cannot receive requests from or allocate resources to an application, as recited in Claim 1, since resources must be

assigned to combinations of multiple clients or computer systems. Appellants respectfully request that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to independent Claim 1 for at least this additional reason.

     Shaha also does not teach or suggest, "determining if the request for resources exceeds the entitlement value associated with the sliding window" as recited in Claim 1. Appellants submit that Shaha does not teach or suggest this limitation as the Examiner has indicated in the Final Office Action dated May 2, 2008. (Final Office Action May 2, 2008, page 3, lines 7-8) First, Shaha does not receive a request for resources but operates by "discovering a utilization pattern of host-service resources from clients by monitoring the clients access rates of one or more websites" (Shaha at Col. 2, lines 59-63; Col. 3, lines 16-18; Col. 3 lines 58-61; Col. 4, lines 12-16; Col. 6, lines 10-15; Col. 7, lines 40-42) Accordingly, it is not possible for Shaha to make any determination with respect to a request for resources as measurements are made after the resources have been accessed and used.

     The section of Shaha at Col. 7, lines 47-49 referenced by the Examiner does not check if a request "exceeds the entitlement value associated with the sliding window" as also recited in Claim 1. Rather, Shaha only states that "the decision support system 116 provides suggestions to a module 122 for allocating resources optimally to clients" based upon a service level agreement (SLA) and prior access patterns of an application. (Shaha at Col. 7, lines 47-57) The SLA in Shaha is not an entitlement value but a contractual agreement describing the costs associated with executing certain operations. Values in the SLA of Shaha specify a cost but do not entitle an application in Shaha to greater or fewer resources.

The combination of Shaha with Berger also does not teach or suggest, "indicating application entitlement to the request for resources in response to the determining and if the request is excessive including a throttling of the requested resources when the application is not entitled to the additional resources in accordance with the entitlement profile." Once again, Shaha does not receive requests for resources and therefore cannot indicate "application entitlement to the request for resources" as recited in Claim 1. Second, Shaha is limited to working with combinations of clients and does not make a determination of whether one application or another is entitled to any resources. (Shaha at Col. 5, lines 45-63)

With respect to "throttling", the Examiner admits Shaha does not teach or suggest "throttling" but alleges Berger does. (Final Office Action, May 2, 2008, page 3, lines 16-18). The Examiner states that Berger includes "a throttle system which determines whether the request will be satisfied at the rate specified by customer request (Col. 5, lines 17-32)".

Appellants submit that the Examiner's characterization of Berger is not only incorrect but does not teach or suggest the limitations in Claim 1. The rate in Berger is not specified by customer request. In Berger, tokens arrive and depart from the different token banks at preassigned rates associated with the class 'i' and according to the function "rate(i)". (Berger at Col. 3, lines 52-56) The rate(i) is not requested by the client but instead is setup in advance as a guaranteed or contracted admission rate for each class 'i'. (Berger at Col. 6, lines 1-8) Berger defines rate(i) as a deterministic flow rate of tokens and therefore credit associated with a class i. (Berger at Col. 8, lines 14-19) For at least this reason, it is important that clients do not request or change the rate(i) over time

as this would make the system non-deterministic and therefore contradict the express

definitions and organization of Berger. Essentially, the rate(i) is fixed for different

classes in advance. (Berger at Col. 11, lines 39-42) Accordingly, the throttle in Berger

does not operate to determine if the request is satisfied at the rate specified by a customer

request.

Accordingly, Appellant respectfully requests that the Board overturn the 35

U.S.C. § 103(a) rejection of record with respect to independent Claims 1, 15, 29 and 30

for at least these aforementioned reasons.

B. DEPENDENT CLAIMS 2-12 AND 16-26 ARE NONOBVIOUS OVER FIELDS IN VIEW OF SHAHA AND FURTHER IN VIEW OF BERGER.

a. ALL CLAIM LIMITATIONS ARE NOT TAUGHT OR SUGGESTED BY THE CITED ART.

Regarding Claims 2-12 and 16-26, Appellant respectfully submits that the

Examiner has misconstrued both Shaha and Berger. For reasons previously described,

neither Shaha or Berger alone or in combination teach or suggest the limitations as

recited in independent Claims 1 and 15 therefore dependent Claims 2-12 and 16-26 are

not only allowable on their own but also by virtue of their dependence on their respective

independent claims.

Accordingly, Appellant respectfully requests that the Board overturn the 35

U.S.C. § 103(a) rejection of record with respect to Claims 2-12 and 16-26 for at least

these reasons.

C. CLAIMS 14 AND 28 ARE NONOBVIOUS OVER SHAHA AS APPLIED TO INDEPENDENT CLAIMS 1 AND 15, IN VIEW OF BERGER AND FURTHER IN VIEW OF DAN.

a. **ALL CLAIM LIMITATIONS ARE NOT TAUGHT OR SUGGESTED BY THE CITED ART.**

Regarding Claims 14 and 28, Appellant respectfully submits that the Examiner has misconstrued Shaha in view of Berger and further in view of Dan. For reasons previously described, neither Shaha or Berger or Dan alone or in combination teach or suggest the limitations as recited in independent Claims 1 and 15 therefore dependent Claims 14 and 28 are not only allowable on their own but also by virtue of their dependence on their respective independent claims.

Accordingly, Appellant respectfully requests that the Board overturn the 35 U.S.C. § 103(a) rejection of record with respect to Claims 14 and 28 for at least these reasons

VIII. CONCLUSION

Appellant respectfully submits it has demonstrated that the cited references do not teach or suggest each and every element of the pending claims 1-26 and that the rejections under 35 U.S.C. § 103(a) cannot be maintained.

For at least the reasons discussed above, Appellant submits that the pending claims are patentable. Accordingly, Appellant requests that the Board of Appeals reverse the Examiner's decisions regarding claims 1-12, 14-26, and 28-30.

IX. **EVIDENCE APPENDIX**

None.

X. **RELATED PROCEEDINGS APPENDIX**

None.

Respectfully submitted,

Date: 01/05/2008   /Leland Wiesner/_____

Leland Wiesner

Reg. No. 39,424

Attorneys for Appellant

Wiesner and Associates

366 Cambridge Ave.

Palo Alto, CA. 94306

Phone: (650) 853-1113

Fax: (650) 853-1114

CLAIMS APPENDIX A

What is claimed is:

This listing of the claims will replace all prior versions, and listings, of claims in the application:

1. A method of policing resources in a computing utility facility, comprising:

intercepting an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application;

acquiring an entitlement profile associated with the application to determine if application is entitled to requested resources over a time period;

identifying an entitlement value and corresponding sliding window of the time period from the entitlement profile;

determining if the request for resources exceeds the entitlement value associated with the sliding window; and

indicating application entitlement to the request for resources in response to the determining and if the request is excessive including a throttling of the requested resources when the application is not entitled to the additional resources in accordance with the entitlement profile.

2. The method of claim 1 further comprising:

acquiring additional sliding windows and corresponding additional entitlement values to determine if the request for resources exceeds at least one entitlement value and sliding window combination; and

indicating that the application is not entitled to the requested resources when the request exceeds the entitlement value in at least one entitlement value and sliding window combination.

3. The method of claim 1 wherein the entitlement profile associated with the application describes the burstiness of the application over the time period.

4. The method of claim 1 wherein a burst loading factor associated with each sliding window corresponds to the burstiness of the application and identifies a portion of an aggregate entitlement to the resources available to fulfill the request.

5. The method of claim 4 wherein a larger burst loading factor is associated with more bursty applications that may need resources more rapidly compared with a smaller burst loading factor is associated with applications that may not need resources as rapidly.

6. The method of claim 1 wherein the entitlement value is derived from historical trace information collected while the application is using resources.

7. The method of claim 1 wherein the burst loading factor is derived from the historical trace information collected while the application is using resources.

8. The method of claim 3 wherein the resource usage is determined according to an estimated probability mass function.

9. The method of claim 4 wherein the estimated probability mass function further includes a confidence interval corresponding to a sample size used for determining the estimated probability mass function.

10. The method of claim 1 wherein the entitlement value operates as a metric for determining whether an application is entitled to the requested resources.

11. The method of claim 10 wherein the entitlement value for an application is proportional to the burstiness of the application in view of resource usage derived from historical trace data.

12. The method of claim 1 wherein determining if the request for resources exceeds the entitlement value further depends on a confidence interval associated with the entitlement value and the number of sample values used to identify the entitlement value.

13. (Canceled)

14. The method of claim 1 wherein indicating application entitlement includes clawing back resources already allocated to the application when the application has exceeded a time limit for using the allocated resources.

15. An apparatus for policing resources in a computing utility facility, comprising:

> a processor capable of executing instructions;

> a memory containing instructions when executed cause the processor to intercept an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application, acquire an entitlement profile associated with the application to determine if application is entitled to requested resources over a time period, identify an entitlement value and corresponding sliding window of the time period from the entitlement profile, determine if the request for resources exceeds the

entitlement value associated with the sliding window and indicate application entitlement to the request for resources in response to the determining and if the request is excessive including a throttling of the requested resources when the application is not entitled to the additional resources in accordance with the entitlement profile.

16. The apparatus of claim 15 further comprising instructions when executed that,

acquire additional sliding windows and corresponding additional entitlement values to determine if the request for resources exceeds at least one entitlement value and sliding window combination and,

indicate that the application is not entitled to the requested resources when the request exceeds the entitlement value in at least one entitlement value and sliding window combination.

17. The apparatus of claim 15 wherein the entitlement profile associated with the application describes the burstiness of the application over the time period.

18. The apparatus of claim 15 wherein a burst loading factor associated with each sliding window corresponds to the burstiness of the application and identifies a portion of an aggregate entitlement to the resources available to fulfill the request.

19. The apparatus of claim 18 wherein a larger burst loading factor is associated with more bursty applications that may need resources more rapidly compared with a smaller burst loading factor is associated with applications that may not need resources as rapidly.

20. The apparatus of claim 15 wherein the entitlement value is derived from historical trace information collected while the application is using resources.

21. The apparatus of claim 15 wherein the burst loading factor is derived from the historical trace information collected while the application is using resources.

22. The apparatus of claim 17 wherein the resource usage is determined according to an estimated probability mass function.

23. The apparatus of claim 18 wherein the estimated probability mass function further includes a confidence interval corresponding to a sample size used for determining the estimated probability mass function.

24. The apparatus of claim 15 wherein the entitlement value operates as a metric for determining whether an application is entitled to the requested resources.

25. The apparatus of claim 24 wherein the entitlement value for an application is proportional to the burstiness of the application in view of resource usage derived from historical trace data.

26. The apparatus of claim 15 wherein determining if the request for resources exceeds the entitlement value further depends on a confidence interval associated with the entitlement value and the number of sample values used to identify the entitlement value.

27. (Canceled)

28. The apparatus of claim 15 wherein indicating application entitlement further includes instructions when executed that claw back resources already allocated to the application when the application has exceeded a time limit for using the allocated resources.

29. A computer program product for policing resources in a computing utility facility, comprising instructions operable to cause a programmable processor to:

intercept an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application;

acquire an entitlement profile associated with the application to determine if application is entitled to requested resources over a time period;

identify an entitlement value and corresponding sliding window of the time period from the entitlement profile;

determine if the request for resources exceeds the entitlement value associated with the sliding window; and

indicate application entitlement to the request for resources in response to the determining and if the request is excessive including a throttling of the requested resources when the application is not entitled to the additional resources in accordance with the entitlement profile.

30. An apparatus for policing resources in a computing utility facility, comprising:

means for intercepting an advance request for resources from an application admitted to access a pool of resources associated with the computing utility facility and prior to utilization of the pool of resources to execute the application;

means for acquiring an entitlement profile associated with the application to determine if application is entitled to requested resources over a time period;

means for identifying an entitlement value and corresponding sliding window of the time period from the entitlement profile;

means for determining if the request for resources exceeds the entitlement value associated with the sliding window; and

means for indicating application entitlement to the request for resources in response to the determining and if the request is excessive including a throttling of the requested resources when the application is not entitled to the additional resources in accordance with the entitlement profile.